REMARKS

Careful review and examination of the subject application are noted and appreciated. Reconsideration of at least independent claim 7 (not amended), independent claims 1, 2 and 14 is respectfully requested as no new issues are believed to have been raised.

SUPPORT FOR THE CLAIM AMENDMENTS

Claim 15 has been moved into claims 1 and 14. Claim 2 has been rewritten in independent form. Thus, no new matter has been added.

OBJECTION UNDER 35 U.S.C. §132

The objection to the prior amendment under 35 U.S.C. §132 for introducing new matter is respectfully traversed and should be withdrawn.

FIG. 3 of the application "illustrates an operation of the pipelined processor encountering a conditional branch instruction." (Application page 10, lines 1-2.) Block 316 generally illustrates an evaluation of a branch condition in an execute stage of the pipelined processor. FIG. 3 also illustrates that while the execute stage is executing, the R-stage may decode an instruction in block 302, load operands in block 304 and check for a branch instruction in decision block 306. FIG. 3 further illustrates that while the execute stage is evaluating the branch condition in the execute stage, the fetch stage may fetch an instruction in block 300.

Block 316 may be reached from the YES branch of decision block 306, which also reaches the select branch target address function in block 314. FIG. 3 generally illustrates that block 314 feeds the branch target address back to block 300 in the fetch stage. "While the branch target address is being generated and fetched, the execution stage may evaluate the branch condition, as shown in block 316. This evaluation may be performed substantially contemporaneously with the generation of the branch target address." (Application, page 11, lines 5-8.)

While the execute stage is evaluating 316 and the fetch stage is fetching 300, the R-stage may **decode** 302, load operands 304 and check for a branch instruction 306 for some other instruction. "In particular, the MIPS ISA **requires a branch-delay slot to follow a conditional branch instruction**. The branch-delay slot may have a no-operation instruction or an instruction from the current thread that can execute before the branch takes effect." (Application, page 5, lines 17-21, emphasis added.) Therefore, the other instruction may be either a no-operation instruction or an instruction that can be executed before the branch takes effect.

The "decoding a second instruction" claim element added by the prior amendment finds antecedent basis in the instructions in the branch-delay slot as explained above and illustrated in FIG. 3. Therefore, no new matter has been added to the application by the prior amendment. As such, the application is fully compliant with 35 U.S.C. §132 and the objection should be withdrawn.

SPECIFICATION OBJECTION

The objection to the title is respectfully traversed and should be withdrawn. No explanation or rational is provided in the Office Action why the title is allegedly non-descriptive. Therefore, the assertion that the title is not descriptive appears to be merely a conclusory statement. Furthermore, steps A-C of claim 1 cover a simple branch prediction method. Step (D), incorporated from claim 15, covers a misprediction recovery. Therefore, the current title appears to be appropriate. Furthermore, the newly proposed title is incorrect as the claimed invention does not wait for an evaluation of the branch condition before fetching the next instruction. As such, the Examiner is respectfully requested to either (i) explain why the title is allegedly non-descriptive or (ii) withdraw the rejection.

DRAWING OBJECTION

The objection to the drawings is respectfully traversed and should be withdrawn.

Please refer to the detailed response in the Objection Under 35 U.S.C. §132 section above. In summary of the detailed response, FIG. 3 generally illustrates evaluating a branch condition in block 316, fetching an instruction 300 at a branch target address 314 and decoding an instruction 302 following the conditional branch instruction. As such, the decoding a second instruction claim element is illustrated in FIG. 3 and the objection should be withdrawn.

CLAIM OBJECTION

The objection to claims 2, 4 and 17-20 has been obviated in part by appropriate amendment, is respectfully traversed in part and should be withdrawn.

Claim 2 has been amended and now includes multiple steps. Regarding claims 4 and 17-20, MPEP §608.01(j) states that the original claim numbering must be preserved throughout the prosecution and new claims must be numbered consecutively beginning with the number next following the highest numbered claim previously presented. Therefore, the gap between added claims 17-20 and the original claim 13 is proper. MPEP §608.01(j) also states that the claims may be renumbered by the Examiner when the application is ready for allowance. Therefore, claim 4 should not be renumbered during prosecution. As such, the objection to the claims should be withdrawn.

CLAIM REJECTIONS UNDER 35 U.S.C. §112

The rejection of claims 1-6 and 14-16 under 35 U.S.C. §112, first paragraph, for enablement is respectfully traversed and should be withdrawn.

The rejection of claims 1-6, and 14-16 under 35 U.S.C. §112, second paragraph, for written description is respectfully traversed and should be withdrawn.

The rejection of claims 1-6, 9 and 14-16 under 35 U.S.C.§112, second paragraph, for indefiniteness has been obviated in part by appropriate amendment, is respectfully traversed in part and should be withdrawn.

Claim 9 has been amended per the Examiner's suggestion. Regarding claims 1-6 and 14-16, please refer to the detailed response in the Objection Under 35 U.S.C. §132 section above. In summary of the detailed response, decoding a second instruction (in the delay-slot by block 302) while a branch target address is being fetched (by block 300) is definite, described and enabled in FIG. 3 and the associated text in pages 5, 10 and 11. As such, the claimed invention is fully compliant with 35 U.S.C. §112, first paragraph and 35 U.S.C. §112, second paragraph and the rejection should be withdrawn.

CLAIM REJECTIONS UNDER 35 U.S.C. §102

The rejection of claims 1-4, 6-10 and 14-16 under 35 U.S.C. §102(b) as being anticipated by Trauben et al. '130 (hereafter Trauben) has been obviated in part by appropriate amendment, is respectfully traversed in part and should be withdrawn.

Trauben concerns a method and apparatus for grouping multiple instructions, issuing grouped instruction simultaneously, and executing groups instruction in a pipelined processor (Title). In contrast, claim 1 provides (in part) a step for fetching a third instruction stored at a mispredict recovery address adjacent a next address in response to determining not to take a branch. Despite the assertion on page 15 of the Office Action (arguing the former claim 15), Trauben appears to contemplate that an address following a branch instruction is fetched prior to evaluating the branch condition, and thus not in response to determining not to take the branch as presently claimed. In particular, FIG. 10b of Trauben shows that a Delay Instruction (asserted similar to the claimed second instruction on page 7, section b of the Office Action) immediately follows a branch instruction BNE. Therefore, a Sequential 1 instruction, adjacent the Delay Instruction as shown in FIG. 10b of Trauben, would appear to be located at an address similar to the claimed mispredict recovery address. However, the Sequential 1 instruction is fetched starting at time 3 which is before time 5 (asserted to be the time when the branch condition is evaluated per page 8, section c of the Office Action) or time 6 (the time that the branch condition is resolved by the integer function unit per column 18, lines 61-63 of Trauben.) Therefore, Trauben does not disclose or suggest a step for fetching a third instruction stored at a mispredict recovery address adjacent a next address **in response to determining not to take a branch** as presently claimed. Claims 14 provides language similar to claim 1. As such, the claimed invention is fully patentable over the cited reference and the rejection should be withdrawn.

Claim 2 provides (in part) a step for fetching a third instruction stored at a sequential instruction address adjacent a branch target address in response to determining to take a branch. Despite the assertion on page 8 of the Office Action, Trauben appears to contemplate that an instruction adjacent a branch target address is fetched starting **before** a determination to take a branch. In particular, FIG. 10a of Trauben illustrates that a Target 2 instruction adjacent a Target 1 instruction (asserted to be at an address similar to the claimed branch target address on page 7, section a of the Office Action) is fetched starting at time 5. Per column 18, lines 61-63 of Trauben and page 8, section 25 of the Office Action, the branch condition is not resolved until stage e0 at time 6. Thus, fetching of Target 2 starting at time 5 does not appear to be performed in response to determining to take a branch which is not known until a later time 6. Therefore, Trauben does not disclose or suggest a step for fetching a third instruction stored at a sequential instruction address adjacent a branch target address in response to determining to take a branch as presently claimed. As such, claim 2 is fully patentable over the cited reference and the rejection should be withdrawn.

Claim 7 provides (in part) a circuit configured to present (i) a branch target address based on a branch instruction stored at a program counter address in prediction of taking a branch, (ii) a sequential instruction address having a first value adjacent the program counter address and (iii) a mispredict recovery address to a multiplexer substantially simultaneously. Despite the assertions starting on page 11, section 29 of the Office Action, a Sequential Queue 58 and a Branch Target Queue 60 shown in FIG. 3 of Trauben do not appear to provide addresses to a multiplexer 62. In particular, column 8, lines 51-60 of Trauben appear to indicate that the queues 58 and 60 provide **instructions**, not **addresses**, to the multiplexer 62. Furthermore, the multiplexer 62 of Trauben only has two inputs which do not appear to support receiving three addresses substantially simultaneously. Therefore, Trauben does not disclose or suggest a circuit configured to present (i) a branch target address based on a branch instruction stored at a program counter address in prediction of taking a branch, (ii) a sequential instruction address having a first value adjacent the program counter address and (iii) a mispredict recovery address to a multiplexer substantially simultaneously as presently claimed. As such, claim 7 is fully patentable over the cited reference and the rejection should be withdrawn.

Claim 3 provides a step for generating a sequential instruction address based upon a program counter address and a predetermined offset. Trauben does not appear to explicitly discuss sequential addresses being a predetermined offset from a program counter address. Therefore, Trauben does not disclose or suggest a step for generating a sequential instruction address based upon a program counter address and a predetermined offset as presently claimed.

Furthermore, Applicant's representative respectfully traverses the assertion on page 8, section 26 of the Office Action that predetermined offsets are inherent. Inherency requires certainty of results, not mere possibility. See, e.g., *Ethyl Molded Products Co. v. Betts Package, Inc.*, 9 U.S.P.Q. 2d 1001 (E.D.Ky 1988). See also, In re Oelrich, 666 F.2d 578, 581, 212 USPQ 323, 326 (C.C.P.A. 1981). In contrast, column 2, lines 43-45 of Trauben states that some instruction may be variable in length. Therefore, an offset from a program counter address pointing to a current instruction to a next instruction cannot be predetermined where the length of the current instruction may vary from one instruction to another. As such, predetermined offsets are not certain and thus not inherent. Claim 6 provides language similar to claim 3. As such, claims 3 and 6 are fully patentable over the cited reference and the rejection should be withdrawn.

Claim 4 provides a step for generating a misprediction recovery address based upon an exception program counter address and a predetermined offset. Page 9, section 27 of the Office Action reasserts that predetermined offsets are inherent. As argued above, instruction lengths may vary, thus predetermined offsets are not certain and therefore not inherent. Thus, Trauben does not disclose or suggest a step for generating a misprediction recovery address based upon an exception program counter address and a predetermined offset as presently claimed. Claim 6 provides language similar to claim 4. As such, claims 4 and 6 are fully patentable over the cited reference and the rejection should be withdrawn.

Claim 6 provides (in part) a step for fetching a fourth instruction stored at a mispredict recovery address in response to determining to not take a branch. As argued above for claim 2, Trauben does not disclose or suggest fetching an instruction stored at a misprediction recover address in response to determining not to take a branch as presently claimed. As such, claim 6 is fully patentable over the cited reference and the rejection should be withdrawn.

Claim 8 provides a circuit configured to present a sequential instruction address having a second value adjacent a branch target address to a multiplexer in response to determining to take a branch. Despite the assertion on page 13, section 30 of the Office Action, Trauben appears to contemplate that an instruction adjacent a branch target address is fetched starting **before** determining to take a branch. In particular, FIG. 10a of Trauben illustrates that a Target 2 instruction adjacent a Target 1 instruction (asserted to be at an address similar to the claimed branch target address on page 12, section 29bi of the Office Action) is fetched starting at time 5. Per column 18, lines 61-63 of Trauben, and page 8, section 25 of the Office Action, the branch condition is not resolved until stage e0 at time 6. Thus, fetching of Target 2 starting at time 5 does not appear to be performed in response to determining to take a branch which is not known until a later time 6. Therefore, Trauben does not disclose or suggest a circuit configured to present a sequential instruction address having a second value adjacent a branch target address to a multiplexer in response to determining to take a branch as presently claimed. As such, claim 8 is fully patentable over the cited reference and the rejection should be withdrawn.

Claim 16 provides a step for storing a program counter address in a stage of a pipelined processor for at least two cycles. Despite the assertion on page 15, section 35 of the Office

Action, Trauben appears to contemplate holding a program counter address for only one cycle. In particular, column 17, lines 59-63 of Trauben state that "f0" and "f1" are stages, not cycles as suggested in the Office Action. Furthermore, column 18, lines 1-2 of Trauben state that each stage takes ½ clock cycles. Thus, a combination of stages f0 and f1 only take one clock cycle, not at least two cycles as presently claimed. As such, claim 16 is fully patentable over the cited reference and the rejection should be withdrawn.

CLAIM REJECTIONS UNDER 35 U.S.C. §103

The rejection of claims 5, 12, 13 and 17-20 under 35 U.S.C. §103(a) as being unpatentable over Trauben in view of Hennessy's book (hereafter Hennessy) is respectfully traversed and should be withdrawn.

The rejection of claim 11 under 35 U.S.C. §103(a) as being unpatentable over Trauben in view of Eckner '460 is respectfully traversed and should be withdrawn

Regarding claim 5, the Office Action fails to provide clear and particular evidence of motivation to combine the references. In particular, the asserted motivation on page 17, section 38 of the Office Action does not claim to be from Trauben, Hennessy or knowledge generally available to one of ordinary skill in the art as required by MPEP §2142. Therefore, the asserted motivation is merely a conclusory statement and *prima facie* obviousness has not been established. As such, claim 5 is fully patentable over the cited references and the rejection should be withdrawn.

Regarding claim 12, the Office Action fails to provide (i) clear and particular evidence of motivation to combine the references and (ii) evidence of a reasonable expectation of success. In particular, the asserted motivation on page 18, section 39 of the Office Action does not claim to be from Trauben, Hennessy or knowledge generally available to one of ordinary skill in the art as required by MPEP §2142. Therefore, the asserted motivation is merely a conclusory statement and *prima facie* obviousness has not been established.

Regarding a reasonable expectation of success, the Office Action asserts on page 17, section 39c that Hennessy teaches "that a sequential PC is always computed for every instruction." In contrast, column 5, lines 49-55 of Trauben state that multiple instructions may be issued simultaneously. Therefore, a conflict appears to exist between Hennessy's approach of computing a program counter address for every instruction and Trauben's approach of issuing multiple instructions simultaneously. As such, the Examiner is respectfully requested to either (i) provide evidence of a reasonable expectation of success in view of the apparent conflict or (ii) withdraw the rejection for claim 12.

Regarding claim 13, the Office Action fails to provide (i) clear and particular evidence of motivation to combine the references and (ii) evidence of a reasonable expectation of success. In particular, the asserted motivation on page 20, section 40 of the Office Action does not claim to be from Trauben, Hennessy or knowledge generally available to one of ordinary skill in the art as required by MPEP §2142. Therefore, the asserted motivation is merely a conclusory statement and *prima facie* obviousness has not been established.

Regarding a reasonable expectation of success, FIG. 10b of Trauben appears to show a method of handling an exception when taking a branch is mispredicted. The method appears to be not decoding the previously fetched Target 1 instruction. Therefore, no clear utility of the exception program counter from Hennessy appears to exist in the context of Trauben. As such, the Examiner is respectfully requested to either (i) provide evidence of a reasonable expectation of success or (ii) withdraw the rejection for claim 13.

Claim 17 provides an incrementor coupled to a prefetch program counter. In contrast, neither of Trauben and Hennessy appear to explicitly mention an incrementor. Therefore, Trauben and Hennessy, alone or in combination, do not teach or suggest an incrementor coupled to a prefetch program counter as presently claimed.

Furthermore, Applicant's representative respectfully traversed the assertion on page 20, section 41 of the Office Action that an incrementor in inherent. Column 2, lines 43-45 of Trauben state that some instructions are variable in length. Therefore, an offset from a program counter address, pointing toward a current instruction, to a next instruction cannot be a known incremental step since the length of the current instruction may vary from one instruction to another. Thus, an incrementor is not certain and thus not inherent. Claim 19 also provides an incrementor. As such, claims 17 and 19 are fully patentable over the cited references and the rejection should be withdrawn.

Claim 20 provides an exception program counter coupled to a prefetch program counter. In contrast, Trauben does not mention an exception program counter and Hennessy does not appear to couple an exception program counter to a prefetch program counter. Therefore, Trauben and Hennessy, alone of in combination, do not teach or suggest an exception program counter coupled to a prefetch program counter as presently claimed.

Furthermore, Applicant's representative respectfully traverses the assertion on page 21, section 44 of the Office Action that an exception program counter inherently updates from a prefetch program counter. In particular, the Office Action asserts on page 7, section 24b that it is "well known in the art function is that the program counter points to the addresses of instructions that are fetched for execution." If the exception program counter is updated from the prefetch program counter upon a mispredicted branch instruction resulting in an exception, the exception program counter would appear to receive an address from somewhere along the mispredicted branch currently being fetched using the prefetch program counter address. Therefore, the exception program counter would not have an address related to either the branch instruction that caused the mispredict exception or any subsequent instruction along the correct branch making recovery from the misprediction uncertain. As such, an exception program counter receiving an address from a prefetch program counter does not appear to result in a workable solution and thus appears to be non-inherent. The Examiner is respectfully requested to either (i) provide an explanation how loading an address from along a mispredicted branch into an exception program counter allows for recovery from the misprediction or (ii) withdraw the rejection for claim 20.

Regarding claim 11, the Office Action fails to provide clear and particular evidence of motivation to combine the references. In particular, the asserted motivation on page 23 of the Office Action does not claim to be from Trauben, Eckner or knowledge generally available to one of ordinary skill in the art as required by MPEP §2142. Therefore, the asserted motivation is merely a conclusory statement and *prima facie* obviousness has not been established.

Furthermore, the asserted advantage on page 22, section 46d of the Office Action that Eckner teaches saving time flushing a pipeline does not appear to be applicable to Trauben. In particular, FIG. 10b of Trauben illustrates instruction Sequential 1 being decoded before the branch condition is evaluated in stage e0. Therefore, no flushing of the pipeline of Trauben appears to occur when a branch is mispredicted. Since Trauben does not flush the pipeline, it is unclear how modifying or combining Trauben with Eckner will result in saving time flushing the pipeline. Therefore, no motivation appears to exist to combine the references and *prima facie* obviousness has not been established. The Examiner is respectfully requested to either (i) explain how the non-flushing branch misprediction operation of Trauben can be shortened by Eckner or (ii) withdraw the rejection.

Accordingly, the present application is in condition for allowance. Early and favorable action by the Examiner is respectfully solicited.

The Examiner is respectfully invited to call the Applicant's representative at 586-498-0670 should it be deemed beneficial to further advance prosecution of the application.

If any additional fees are due, please charge Deposit Account No. 12-2252.

Respectfully submitted,

CHRISTOPHER P. MAIORANA, P.C.

Christopher P. Maiorana
Registration No. 42,829

Dated: May 21, 2004

c/o Sandeep Jaggi
LSI Logic Corporation
1621 Barber Lane, M/S D-106 Legal
Milpitas, CA 95035

Docket No.: 00-419 / 1496.00055